



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



Desarrollo de API de prueba para el registro y ampliación de clientes en la base de datos de la Compañía Financiera TUYA S. A.



COMPAÑÍA FINANCIERA TUYA S.A

ANDRES FELIPE DEVIA MUÑOZ

UNIVERSIDAD DE CÓRDOBA

FACULTAD DE INGENIERÍAS

INGENIERIA DE SISTEMAS

MONTERIA, CÓRDOBA

2022



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



Desarrollo de API de prueba para el registro y ampliación de clientes en la base de datos de la Compañía Financiera TUYA S. A.



COMPAÑIA FINANCIERA TUYA S.A

ANDRES FELIPE DEVIA MUÑOZ

Trabajo de grado presentado, en la modalidad de Práctica Empresarial para optar al Título de INGENIERO DE SISTEMAS

Director (es):

MILTON EDUARDO HERNANDEZ ZAKZUK, M.Sc.

MARCELA RESTREPO LOPEZ, M.Sc.

UNIVERSIDAD DE CÓRDOBA

FACULTAD DE INGENIERÍAS

INGENIERIA DE SISTEMAS

MONTERIA, CÓRDOBA

2022



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



La responsabilidad ética, legal y científica, de las ideas, conceptos, y resultados del proyecto de investigación, serán responsabilidad de los autores.

Artículo 59, Acuerdo N° 022 del 21 de febrero de 2018 del Consejo Superior.

Tener en cuenta los Artículos y directrices establecidos la Resolución 1775, del 21 de agosto de 2019. En donde se establecen las directrices y las políticas de funcionamiento del repositorio institucional de la Universidad de Córdoba (Artículos tercero, octavo, once, entre otros).

“11 – BUENA FE: La universidad considera que la producción intelectual que, los profesores, funcionarios administrativos y estudiantes le presenten, es realizada por éstos, y que no han transgredido los derechos de otras personas. En consecuencia la aceptará, protegerá, publicará y explotará, según corresponda y lo considere pertinente”. Artículo 1, Acuerdo N° 045 del 25 de mayo de 2018 del Consejo Superior.



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.
INFORME FINAL PRÁCTICA EMPRESARIAL
Comité de Acreditación y Currículo Facultad de Ingenierías



Nota de aceptación

Firma del jurado

Firma del jurado





UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.
INFORME FINAL PRÁCTICA EMPRESARIAL
Comité de Acreditación y Currículo Facultad de Ingenierías



*A mis padres Janet Muñoz Hernández y
Luis Miguel Devia Hurtado*

A mi hermano Miguel Ángel Devia Muñoz

*A mi novia Dalmis Manuela Paternina
Mejía*

A mis familiares

*A mis amigos, en especial a Osmel De Jesús
Pérez Gonzales*



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



TABLA DE CONTENIDO

| | |
|--|----|
| 1. INTRODUCCIÓN | 10 |
| 2. PLATAFORMA ESTRATÉGICA DE LA EMPRESA | 11 |
| 2.1. Sueño de la compañía TUYA S.A. | 12 |
| 2.2. Aspiración de la Compañía TUYA S.A. | 13 |
| 2.3. Valores de la Compañía TUYA S.A. | 13 |
| 2.4. Estructura Organizacional de la compañía..... | 14 |
| 3. MARCO TEÓRICO O CONCEPTUAL..... | 14 |
| 3.1. ¿Qué es una API?..... | 14 |
| 3.2. Metodología Scrum | 15 |
| 3.3. Controlador Web API | 16 |
| 3.4. Métodos de petición HTTP | 16 |
| 3.5. Método GET | 16 |
| 3.6. Método POST | 16 |
| 3.7. Método PUT | 17 |
| 3.8. Método DELETE | 17 |
| 3.9. Método PATCH | 17 |
| 3.1.1 Base de datos | 17 |
| 3.1.2 Front End | 17 |
| 3.1.3 Back End | 18 |
| 4. DIAGNÓSTICO ESPECÍFICO DE LA EMPRESA | |
| 5. ACTIVIDADES DESARROLLADAS..... | |



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



| | |
|---|----|
| 5.1. Adaptación al puesto..... | 19 |
| 5.2. Reuniones de planeación | 28 |
| 5.3. Creación de la base de datos en SQL Server | 31 |
| 5.4. Creación de la tabla Clientes en la base de datos..... | 31 |
| 5.5. Creación de la Entidad principal de nuestra API | 32 |
| 5.6. Creación de las funcionalidades requeridas en la API. | 34 |
| 5.7. Clase para controlar todas las funciones de la API | 45 |
| 5.8. Reporte final sobre el estado de la API..... | 50 |
| 5.9. Ejecución de la API. | 52 |
| 6. PROPUESTA DE MEJORAMIENTO | 53 |
| 7. APORTES DEL ESTUDIANTE..... | 53 |
| 8. CONCLUSIONES | 54 |
| 9. RECOMENDACIONES | 55 |
| 10. BIBLIOGRAFÍA..... | 56 |



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



LISTADO DE GRÁFICOS

| | |
|--|----|
| Gráfico 1. Valores de la Compañía TUYA S.A..... | 13 |
| Gráfico 2. Estructura Organizacional de la Compañía TUYA S.A. | 14 |
| Gráfico 3. Curso Cumplimiento | 20 |
| Gráfico 4. Curso Fundamentos de Agilismo | 21 |
| Gráfico 5. Curso Monitoreo Application Insights Venta Online | 22 |
| Gráfico 6. Curso Lineamiento de seguridad para el registro de eventos..... | 23 |
| Gráfico 7. Manual de referencia creación de Dashboards en Application Insights | 24 |
| Gráfico 8. Manual de referencia KQL para Logs de Application Insights | 25 |
| Gráfico 9. Procedimiento para crear y configurar alertas con ayuda de Application Insights | 26 |
| Gráfico 10. Archivos alojados en la Intranet de la compañía..... | 27 |
| Gráfico 11. Comunidad de practica de la compañía | 28 |
| Gráfico 12. Requisitos técnicos para la API a desarrollar..... | 29 |
| Gráfico 13. Conocimientos aplicados para el desarrollo de la API..... | 29 |
| Gráfico 14. Requisitos para el reto planeado | 30 |
| Gráfico 15. Requisitos para el reto planeado parte 2 | 30 |
| Gráfico 16. Creación de la base de datos ClientesTuyaDb en SQL Server | 31 |
| Gráfico 17. Creación de la tabla Clientes en la base de datos ClientesTuyaDb..... | 32 |
| Gráfico 18. Creación de la entidad principal de nuestra API..... | 33 |
| Gráfico 19. Creación de la entidad principal de nuestra API parte 2 | 34 |
| Gráfico 20. Interfaz IRegistrarCliente para insertar cliente nuevo..... | |
| Gráfico 21. Clase ClienteCreacionDTO donde se crea un cliente nuevo..... | |



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



| | |
|---|----|
| Gráfico 22. Clase ClienteCreacionDTO donde se crea un cliente nuevo parte 2..... | 37 |
| Gráfico 23. Interfaz IListarDoc para buscar un cliente por su número de documento..... | 38 |
| Gráfico 24. Clase ClienteListarDocDTO donde se lista un cliente por su número de documento .. | 39 |
| Gráfico 25. Clase ClienteListarDocDTO donde se lista un cliente por su número de documento parte 2..... | 40 |
| Gráfico 26. Interfaz IActualizarEstado para inactivar cliente por su número de documento | 41 |
| Gráfico 27. Clase ClienteUpdateEstadoDTO para inactivar un cliente por su número de documento | 42 |
| Gráfico 28. Actualización de los datos del cliente por su número de documento por medio la interfaz IActualizarCliente.cs | 43 |
| Gráfico 29. Actualización de los datos de un cliente por su número de documento con la clase ClienteUpdateDTO.cs | 44 |
| Gráfico 30. Actualización de los datos de un cliente por su número de documento con la clase ClienteUpdateDTO.cs parte 2 | 45 |
| Gráfico 31. Controlador de las funciones de la API..... | 46 |
| Gráfico 32. Método Post para registrar clientes a la base de datos | 47 |
| Gráfico 33. Método Get para buscar un cliente en la base de datos..... | 48 |
| Gráfico 34. Método Put para cambiar el estado de un cliente en la base de datos | 49 |
| Gráfico 35. Método Put para actualizar los datos de un cliente en la base de datos | 50 |
| Gráfico 36. Reunión de reporte final con la líder Marcela Restrepo López..... | 51 |
| Gráfico 37. Ejecución de la API..... | 52 |



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



1. INTRODUCCIÓN

TUYA S.A. es una compañía de financiamiento que pertenece a Grupo Bancolombia y Grupo ÉXITO, se trata del segundo emisor de tarjetas en Colombia con 2.850.000 tarjetas emitidas activas en la actualidad, tiene 2,841 empleados y alcanza una facturación superior a los 2,7 billones de pesos. TUYA se define como una compañía que promueve la financiación responsable del consumo, a través de diversos productos de crédito que son puestos a disposición de los colombianos en alianza con almacenes de cadena como Éxito y Alkosto (SAS, S.f.). Esta lanzó al mercado la billetera digital TuyaPay, primer producto transaccional y de apertura 100% digital, que a la fecha cuenta con más de 805 mil usuarios.

La Compañía Financiera en su aérea de ventas online está encargada de la venta en línea de sus productos de tarjetas de crédito y créditos no rotativos; por esta razón buscan la creación de estrategias para la implementación de nuevas funcionalidades tecnológicas o en su defecto automatizar o mejorar las ya creadas por la compañía, esto para mejorar la calidad de sus servicios implicados con su página de ventas. Hoy, gracias a la diferenciación de su oferta, esta empresa tiene un mayor flujo de clientes a los cuales se les debe brindar la mejor experiencia, por lo que se hace necesario realizar pilotos de registro de clientes y esto define el objetivo a cumplir durante el desarrollo de las practicas, el cual es: construir una API de prueba para el registro y la ampliación de clientes en la base de datos de la compañía financiera TUYA S.A.

Además de la creación de estas nuevas funcionalidades también se debe dar soporte y solución a incidentes asociados a todo el desarrollo realizado durante el tiempo activo como empleado practicante de la empresa, todo esto con el fin de aprender e impulsar de manera fundamental el crecimiento y fortalecimiento de la dependencia.



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



2. PLATAFORMA ESTRATÉGICA DE LA EMPRESA

La Compañía TUYA S.A antes de ser lo que es hoy era una sociedad constituida bajo la denominación COLOMBO MEXICANA DE INVERSIONES S.A – COLMEX, mediante escritura pública No. 7418 de noviembre 30 de 1971, otorgada en la Notaria Primera del Círculo de Bogotá, D.C.

En 1980, la Compañía fue adquirida por el Grupo Suramericana y modifica su domicilio a la ciudad de Medellín y su denominación social, pasando a ser COMPAÑÍA SURAMERICANA DE FINANCIAMIENTO COMERCIAL S.A. “SUFINANCIAMIENTO”. En la década de los años 80, la Compañía estaba orientada a satisfacer necesidades de crédito de la pequeña y mediana empresa, estrategia que cambió en 1992 con el desarrollo de líneas de crédito dirigidas a personas naturales, como tarjetas de crédito de marca privada, créditos de vehículo y libre inversión entre otros, los cuales permitieron el posicionamiento de la Compañía en el mercado del crédito de consumo.

En el año 2003, la Compañía fue adquirida por Bancolombia S.A, Colcorp Corporación Financiera S.A., Fondo de Empleados de Bancolombia e Inmobiliaria Bancol S.A, pasando a formar parte del Grupo Bancolombia.

En el primer trimestre del año 2010, con ocasión de la celebración del contrato de cesión de activos, pasivos y contratos con Bancolombia S.A., la Compañía cambió su denominación social pasando a ser COMPAÑÍA DE FINANCIAMIENTO TUYA S.A.

El 31 de octubre de 2016, y una vez obtenidas las autorizaciones regulatorias correspondientes por parte de la superintendencia financiera de Colombia, Bancolombia S.A., el Fondo de Empleados del Grupo Bancolombia - FEBANC y la fundación Bancolombia (vendedores), transfirieron a almacenes Éxito S.A. y almacenes Éxito



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



Inversiones S.A.S. (compradores) el 50% de las acciones en circulación de la Compañía de Financiamiento TUYA S.A. En el marco de esta transacción se celebraron las siguientes operaciones:

- Almacenes Éxito S.A. y su vinculada almacenes Éxito Inversiones S.A.S., pasan a ser titulares del 50% de las acciones de TUYA.
- El 50% de las acciones de TUYA en cabeza del Grupo Bancolombia, continuarán a través de Bancolombia y de sus subsidiarias Banca de Inversión Bancolombia S.A. Corporación Financiera y de BIBA Inmobiliaria S.A.S. (Actualmente Pasarela Colombia S.A.S.). (Tuya 2022)

2.1. Sueño de la compañía TUYA S.A.

En Tuya, creemos en ti. Por eso somos la mano amiga que te da oportunidades para que puedas decirles sí a tus sueños. Nos mueven las personas como tú, ¡valientes! que se levantan a diario con la confianza puesta en que sí se puede, en que es posible dar un paso adelante para transformar sus vidas, para hacer crecer sus negocios. Por eso abrimos las puertas a las oportunidades. Queremos estar ahí, acompañándote para que tomes decisiones con la seguridad y tranquilidad de que lo que sueñas puede ser una realidad, sin importar qué tan pequeño o grande sea. Creemos en que juntos haremos del mundo un lugar mejor y que al ser conscientes de nuestras acciones, impactamos el planeta y la sociedad de forma positiva. ¡En Tuya soñamos con dar oportunidades que transforman vidas positivamente! (Tuya 2022)



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.
INFORME FINAL PRÁCTICA EMPRESARIAL
Comité de Acreditación y Currículo Facultad de Ingenierías



2.2. Aspiración de la Compañía TUYA S.A.

Ser el mejor aliado financiero a través del Banking As A Service, habilitando soluciones rentables y sostenibles que contribuyan a transformar el día a día de nuestros clientes y empleados y de esta forma construir un mejor país. (Tuya 2022)

2.3. Valores de la Compañía TUYA S.A.



Íntegros



Conscientes



Autónomos



Flexibles



Innovadores

Gráfico 1. Valores de la Compañía TUYA S.A

Tomada de: (Tuya 2022)



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co

2.4. Estructura Organizacional de la compañía.

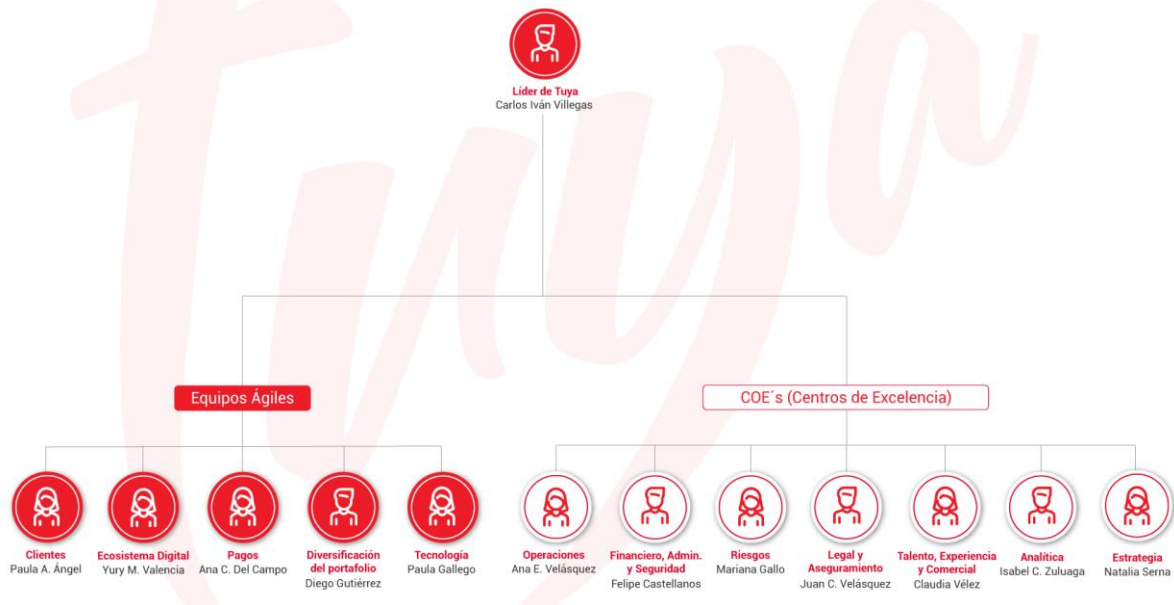


Gráfico 2. Estructura Organizacional de la Compañía TUYA S.A.

Tomada de: (Tuya 2022)

3. MARCO TEÓRICO O CONCEPTUAL

3.1. ¿Qué es una API?

Las API son traductores cuya función es conectar sistemas, software y aplicaciones. Con las API es posible ofrecerle una experiencia de uso más familiar a las personas. El propósito de una API es intercambiar datos entre diferentes sistemas, la mayoría de las veces estos intercambios de datos tienen como objetivo automatizar procesos manuales y/o permitir la creación de nuevas funcionalidades. Las API se pueden utilizar de diferentes formas: integrando diferentes sistemas para una mayor eficiencia a la hora de utilizarlas, y tienen un papel estratégico en la rutina de las empresas.



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



Después de todo, hay varios sistemas y aplicaciones que se utilizan en una empresa, y todas estas funciones interactúan con otro software, a través de la API. (Cardozo 2021)

Las API surgieron casi junto a la informática, incluso antes que la computadora personal. Para ese tiempo, una API usualmente se usaba como biblioteca para los sistemas operativos. Normalmente estaban habilitadas localmente en los sistemas en los que operaban, aunque a veces pasaban mensajes entre las computadoras centrales. Después de casi 30 años, las API se expandieron más allá de los entornos locales. A principios del año 2000, ya eran una tecnología importante para la integración remota de datos. (Red Hat 2017)

3.2. Metodología Scrum

Scrum se basa en el empirismo y el pensamiento Lean. El empirismo afirma que el conocimiento proviene de la experiencia y la toma de decisiones basadas en lo que se observa. El pensamiento Lean reduce los desperdicios y se centra en lo esencial. Scrum emplea un enfoque iterativo e incremental para optimizar la previsibilidad y controlar el riesgo. Scrum involucra a grupos de personas que colectivamente tienen todas las habilidades y experiencia para hacer el trabajo y compartir o adquirir tales habilidades según sea necesario. Scrum combina cuatro eventos formales para la inspección y adaptación dentro de un evento contenedor, el Sprint. Estos eventos funcionan porque implementan los pilares empíricos de Scrum: transparencia, inspección y adaptación. Scrum es un marco ligero que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptables para problemas complejos. (Sutherland, 2020)



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



3.3. Controlador Web API

Un controlador Web API es un controlador de nuestra aplicación que nos permite utilizar sus métodos y acciones vía AJAX, esto nos permite lograr construir secciones de nuestra aplicación que podemos reutilizar en diferentes contextos. Esta herramienta nos ayuda entonces a reducir la cantidad de código que construimos, haciendo que podamos programar algo que es más fácil de mantener y depurar, brindándonos un proyecto más limpio y moderno. (Ferrara, 2014)

3.4. Métodos de petición HTTP

HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados HTTP verbs. Cada uno de ellos implementan una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos. Los más utilizados son: GET, POST, PUT, DELETE, PATCH. (Mozilla, 2021)

3.5. Método GET

El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos. (Mozilla, 2021)

3.6. Método POST

El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor. (Mozilla, 2021)



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11. INFORME FINAL PRÁCTICA EMPRESARIAL Comité de Acreditación y Currículo Facultad de Ingenierías



3.7. Método PUT

El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición. (Mozilla, 2021)

3.8. Método DELETE

El método DELETE borra un recurso en específico. (Mozilla, 2021)

3.9. Método PATCH

El método PATCH es utilizado para aplicar modificaciones parciales a un recurso. (Mozilla, 2021)

3.1.1 Base de datos

Una base de datos es una colección de archivos relacionados que permite el manejo de la información de alguna compañía. Cada uno de dichos archivos puede ser visto como una colección de registros y cada registro está compuesto de una colección de campos. (Cruz-Chávez, 2011).

3.1.2 Front End

Front End es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas (RWD), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales... y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario. (Stefaniak, 2019).



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



3.1.3 Back End

El back end del sitio web consiste en un servidor, una aplicación y una base de datos. Se toman los datos, se procesa la información y se envía al usuario. Los desarrolladores de Front end y Back end suelen trabajar juntos para que todo funcione correctamente. (Stefaniak, 2019)

4. DIAGNÓSTICO ESPECÍFICO DE LA EMPRESA

Una compañía de ventas siempre quiere estar atrayendo nuevos clientes, por lo tanto, tiene que estar en busca de mejoras día a día, este es el caso de la Compañía TUYA especialmente en el departamento de Ventas Online.

En este sentido la empresa en busca de mejorar nos da a conocer la necesidad de dar solución al mejoramiento de los procesos de registro de clientes activos y nuevos a través de dicha API, bajo la temática del modelo de trabajo Scrum, para luego evaluar el efecto de la misma.

Este trabajo permitirá mostrar que mediante la implementación de una API el manejo de los datos de usuarios sea más ágil y seguro, además de dar la facilidad de crear nuevos servicios o funciones que la empresa realice a futuro.

Aquí es donde se intenta ayudar a cumplir esa meta de mejorar, escuchando las necesidades del cliente para luego plasmarlas en los servicios brindados, además buscando la innovación de nuevas funcionalidades mediante una API que le ayude al a página web encargada de las ventas a distribuir el flujo de peticiones, ya sea de registros a la compañía o consultas de registros ya existentes, con el objetivo de facilitar en lo



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



máximo los procesos de filtros manejados por la compañía para el consumo de los productos brindados que esta ofrece.

Así como buscamos la innovación también se busca agilizar las funcionalidades que ya viene implementando la compañía, esto lo hacemos ayudando con la implementación de una API que va ayudar a automatizar estas funcionalidades, con la intención de evitar que el cliente tenga que gastar demasiado tiempo en consumir los servicios brindados por la página web.

5. ACTIVIDADES DESARROLLADAS

5.1. Adaptación al puesto

En esta etapa se definió cuales serian mis funciones dentro de la empresa y las funciones dentro del equipo de trabajo, recibí explicación sobre los trabajos previos realizados en el aérea y realicé cursos sobre la metodología manejada por la compañía.

Estos cursos fueron:



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.
INFORME FINAL PRÁCTICA EMPRESARIAL
Comité de Acreditación y Currículo Facultad de Ingenierías



➤ Curso Cumplimiento



Gráfico 3. Curso Cumplimiento

Tomada de: Hecho por el practicante, 2022





UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.
INFORME FINAL PRÁCTICA EMPRESARIAL
Comité de Acreditación y Currículo Facultad de Ingenierías



➤ **Curso Fundamentos de Agilismo**



Gráfico 4. Curso Fundamentos de Agilismo

Tomada de: Hecho por el practicante, 2022



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co

➤ Curso Monitoreo Application Insights Venta Online

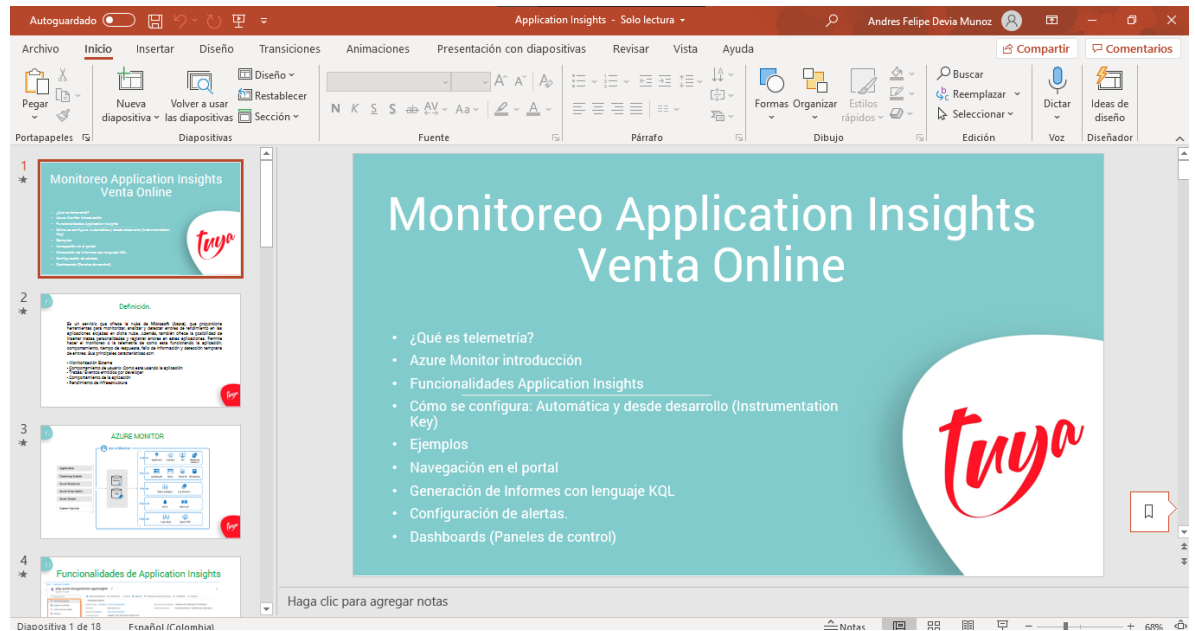
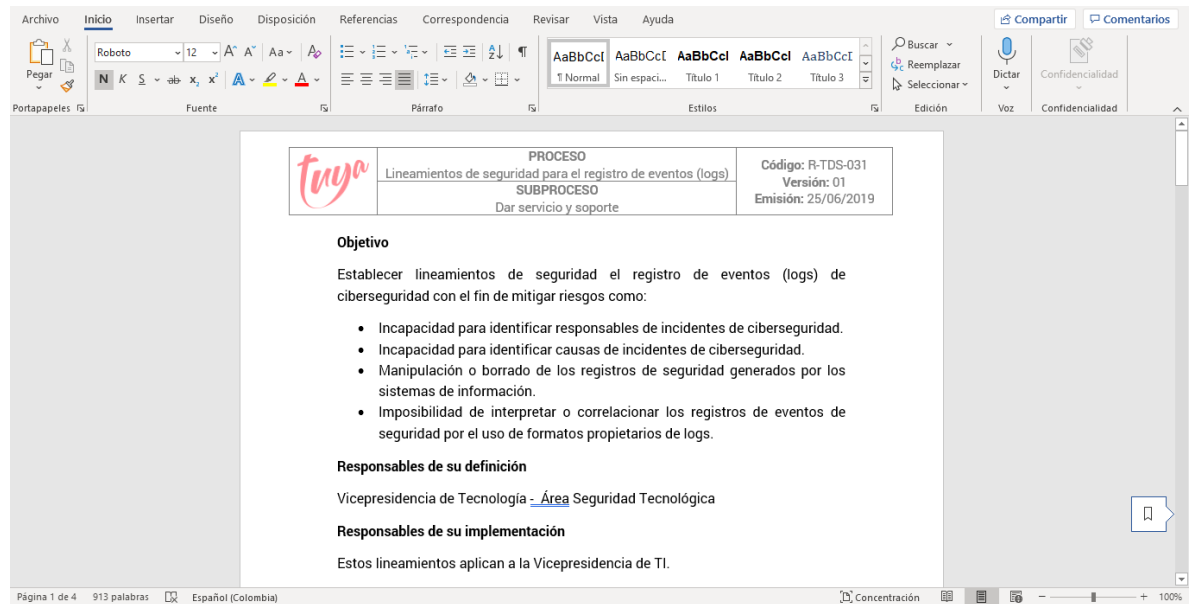



Gráfico 5. Curso Monitoreo Application Insights Venta Online

Tomada de: Hecho por el practicante, 2022

➤ Curso Lineamiento de seguridad para el registro de eventos



The screenshot shows a Microsoft Word document template. The header section contains a table with the following information:

| | | |
|---|--|---------------------|
|  | PROCESO | Código: R-TDS-031 |
| | Lineamientos de seguridad para el registro de eventos (logs) | Versión: 01 |
| | SUBPROCESO | Emisión: 25/06/2019 |
| | Dar servicio y soporte | |

Objetivo

Establecer lineamientos de seguridad el registro de eventos (logs) de ciberseguridad con el fin de mitigar riesgos como:

- Incapacidad para identificar responsables de incidentes de ciberseguridad.
- Incapacidad para identificar causas de incidentes de ciberseguridad.
- Manipulación o borrado de los registros de seguridad generados por los sistemas de información.
- Imposibilidad de interpretar o correlacionar los registros de eventos de seguridad por el uso de formatos propietarios de logs.

Responsables de su definición

Vicepresidencia de Tecnología - Área Seguridad Tecnológica

Responsables de su implementación

Estos lineamientos aplican a la Vicepresidencia de TI.

Gráfico 6. Curso Lineamiento de seguridad para el registro de eventos

Tomada de: Hecho por el practicante, 2022

Además de dichos cursos, también se realizaron lecturas sobre dos distintos manuales y procesos de la compañía. Estos fueron:

➤ **Manual de referencia creación de Dashboards en Application Insights**

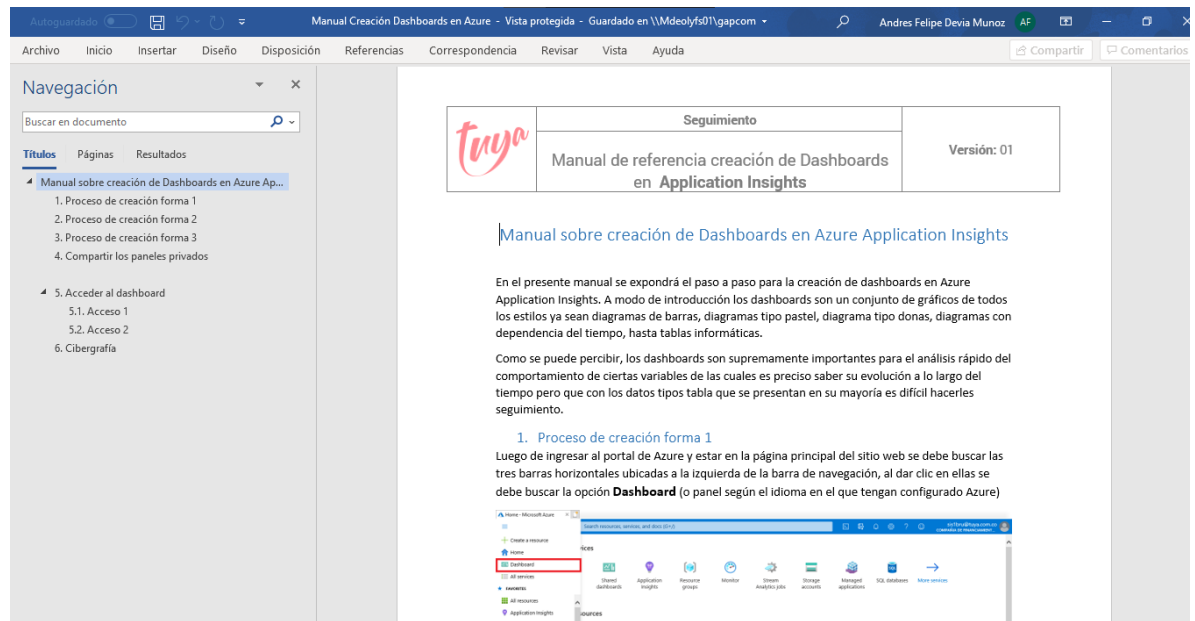


Gráfico 7. Manual de referencia creación de Dashboards en Application Insights

Tomada de: Hecho por el practicante, 2022

➤ Manual de referencia KQL para Logs de Application Insights

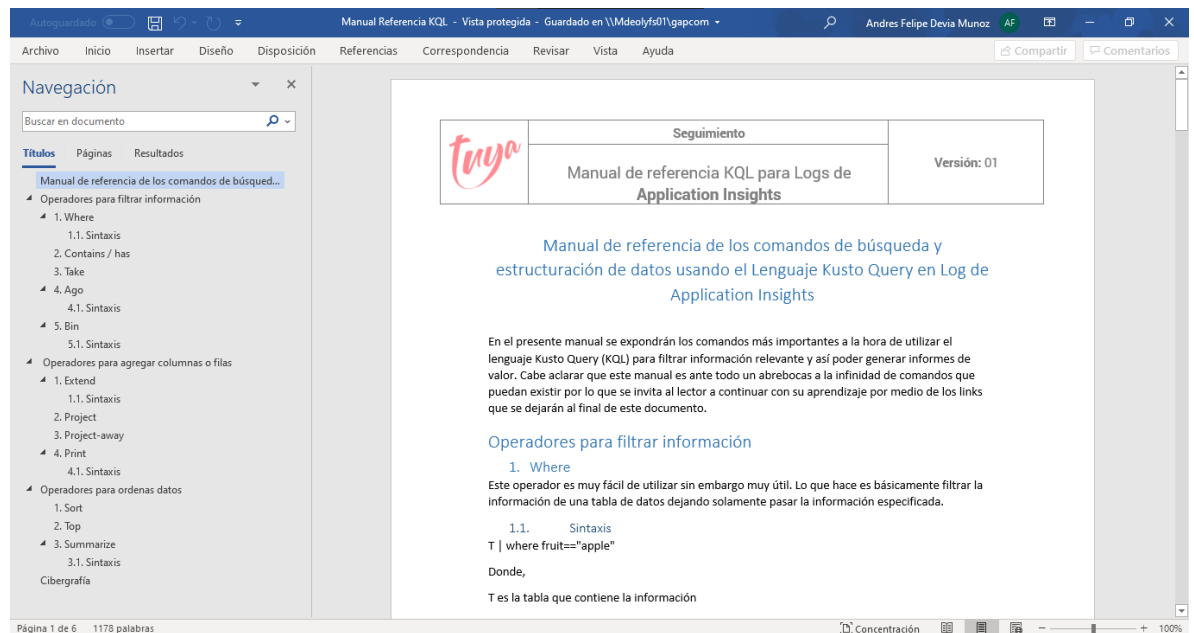
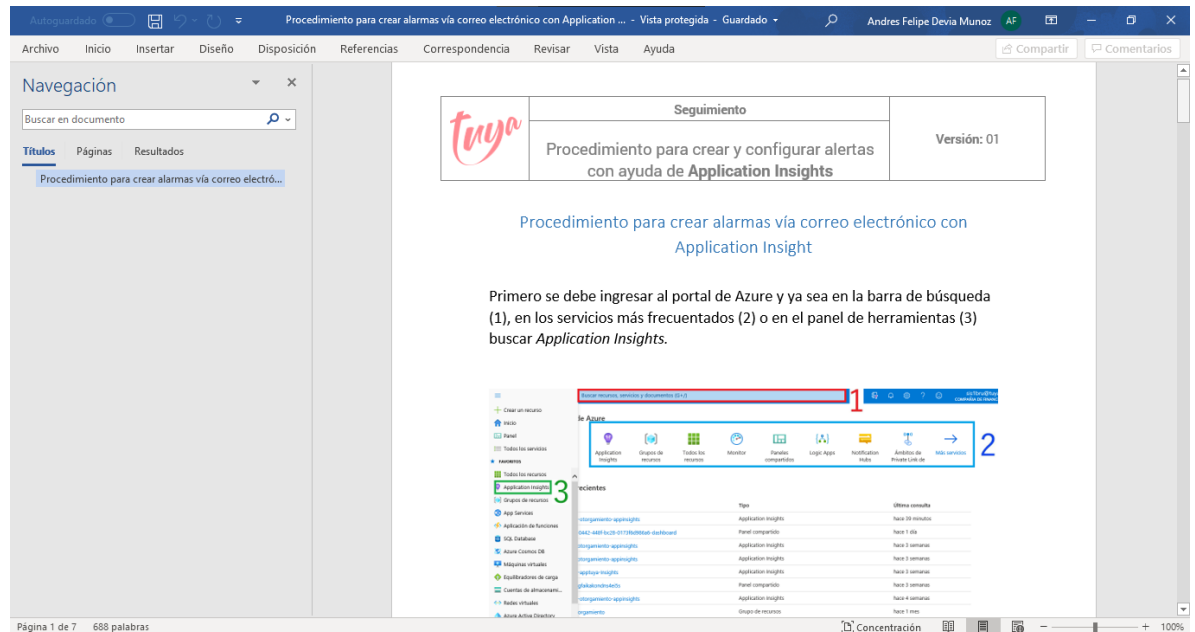


Gráfico 8. Manual de referencia KQL para Logs de Application Insights

Tomada de: Hecho por el practicante, 2022

➤ Procedimiento para crear y configurar alertas con ayuda de Application Insights



The screenshot shows a document with the following content:

Seguimiento
Procedimiento para crear y configurar alertas con ayuda de Application Insights
Versión: 01

Procedimiento para crear alarmas vía correo electrónico con Application Insights

Primero se debe ingresar al portal de Azure y ya sea en la barra de búsqueda (1), en los servicios más frecuentados (2) o en el panel de herramientas (3) buscar Application Insights.

The screenshot also shows a table of recent resources:

| Nombre | Tipo | Última consulta |
|-------------------------------|----------------------|-----------------|
| integracion-appinsights | Application Insights | Hace 10 minutos |
| 0442-4442-0442-0442-0442-0442 | Panel compartido | Hace 1 día |
| integracion-appinsights | Application Insights | Hace 3 semanas |
| integracion-appinsights | Application Insights | Hace 3 semanas |
| integracion-appinsights | Application Insights | Hace 3 semanas |
| integracion-appinsights | Panel compartido | Hace 3 semanas |
| integracion-appinsights | Application Insights | Hace 4 semanas |
| integracion-appinsights | Application Insights | Hace 1 mes |

Gráfico 9. Procedimiento para crear y configurar alertas con ayuda de Application Insights

Tomada de: Hecho por el practicante, 2022

➤ A demás, se hizo la lectura de archivos alojados en la Intranet de la compañía

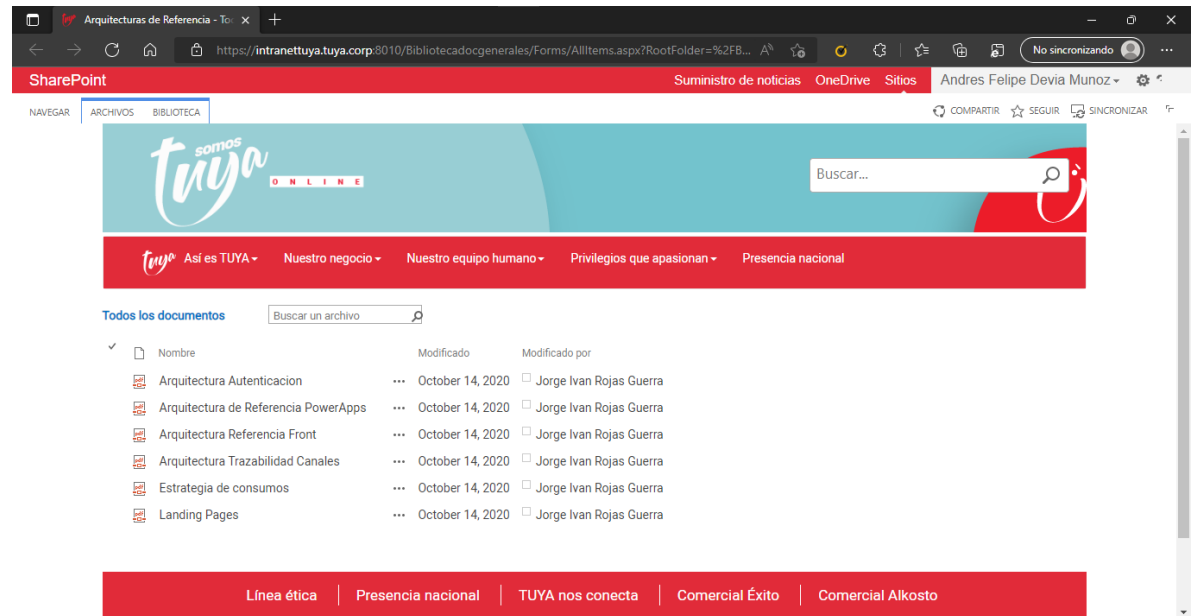


Gráfico 10. Archivos alojados en la Intranet de la compañía

Tomada de: Hecho por el practicante, 2022

Por último, quiero resaltar que la compañía tiene una comunidad de practica para dar apoyo con reuniones semanales con distintos temas, con el objetivo de fortalecer los conocimientos y así facilitando las tareas asignadas.

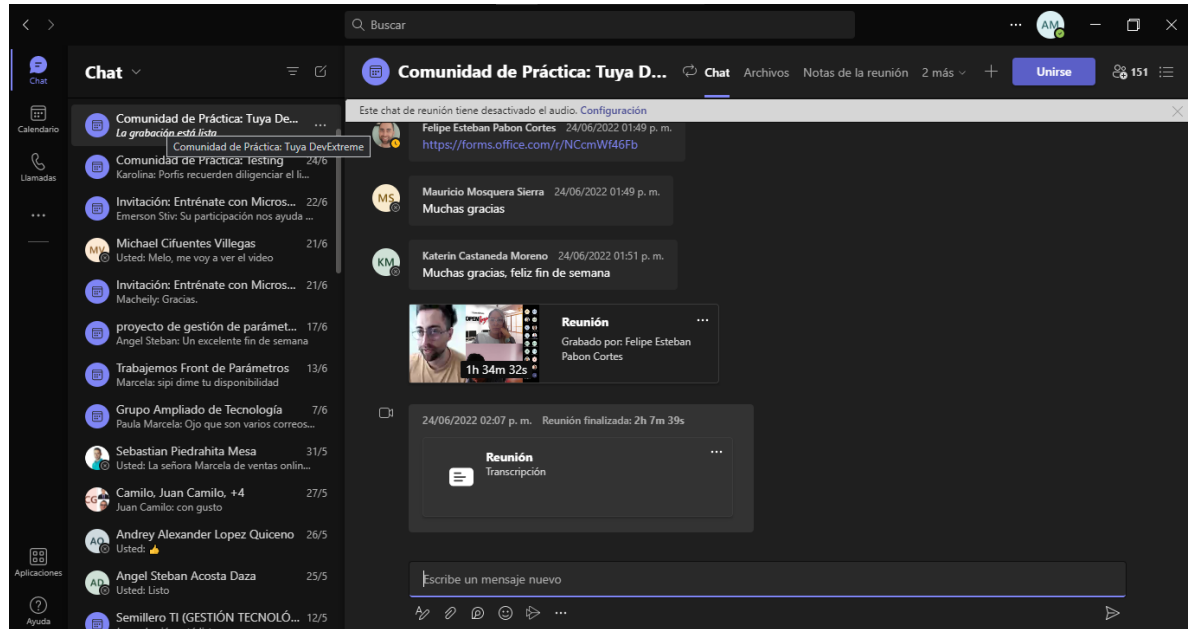


Gráfico 11. Comunidad de practica de la compañía

Tomada de: Hecho por el practicante, 2022

5.2. Reuniones de planeación

Por medio de reuniones virtuales en la plataforma Teams con la líder de equipo Marcela Restrepo López, se habló sobre los requerimientos que debía contener la API a crear.

Dichos requerimientos fueron:



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11. INFORME FINAL PRÁCTICA EMPRESARIAL Comité de Acreditación y Currículo Facultad de Ingenierías



- **Requisitos Técnicos:**

Examen Desarrollador Pleno .NET Core

tuya

Requisitos técnicos

- Acceso libre a internet.
- Visual Studio 2019 – Visual Studio Code
- .NET Core 3.1 o Superior
- Motor Base de Datos Local SQL SERVER
- Crear su proyecto en Documents\PruebaNetCore\Mantenimiento
- Almacenar sus Scripts SQL en Documents\PruebaNetCore\SQL (Solo si tiene scripts)

Gráfico 12. Requisitos técnicos para la API a desarrollar

Tomada de: Hecho por el practicante, 2022

- **Conocimientos aplicados:**

Conocimientos aplicados:

- Desarrollo C# en .NET Core
- Desarrollo EF en .NET Core
- Conocimiento Patrones de Arquitectura de Software.
- Manejo de Base de Datos Estructuradas SQL SERVER.
- Buenas Practicas de Desarrollo.

Gráfico 13. Conocimientos aplicados para el desarrollo de la API

Tomada de: Hecho por el practicante, 2022



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías

- **Reto planeado**

Reto planteado.

La Compañía de Financiamiento TUYA S.A, desea desarrollar para su centro de Contactos una aplicación que permita realizar el mantenimiento de sus clientes.

Requisitos:

- Desarrollar una interfaz(Pantalla) para realizar el ingreso y mantenimiento de Clientes.
- Las acciones mínimas de mantenimiento son:
 - Insertar Datos Principales del cliente (según la entidad del numeral d)
 - Listar todos los Clientes Activos.
 - Listar Clientes por su número de Documento.
 - Inactivar Cliente por su número de Documento
 - Actualizar Datos Personales Cliente por su número de Documento.
- El mantenimiento debe realizarse mediante una API REST hacia la base de datos.
- Los datos deben estar presentes en una base de datos Local SQLSERVER (ClientesTuyaDB)

Gráfico 14. Requisitos para el reto planeado

Tomada de: Hecho por el practicante, 2022

- La entidad requerida para el cliente debe poseer:
 1. Numero de Documento del cliente (PK).
 2. Nombre Completo del cliente.
 3. Correo Electrónico del cliente.
 4. Número Telefónico del cliente.
 5. Dirección del cliente.
 6. Ciudad del Cliente.
 7. Estado Cliente
 - (0) Activo – Por defecto
 - (1) Inactivo
 8. Fecha Creación Cliente
 9. Fecha Inactivación Cliente
 10. Fecha Actualización Cliente
- e. La API REST debe estar desarrolladas en .NET Core Superior a 3.1
- f. Se debe implementar un desarrollo por Capas (MVC) tanto para la interfaz como la API REST.

Gráfico 15. Requisitos para el reto planeado parte 2

Tomada de: Hecho por el practicante, 2022

5.3. Creación de la base de datos en SQL Server

Esta tarea consiste en la creación de la base de datos llamada ClientesTuyaDb, donde se guardarán todos los registros de los nuevos clientes.

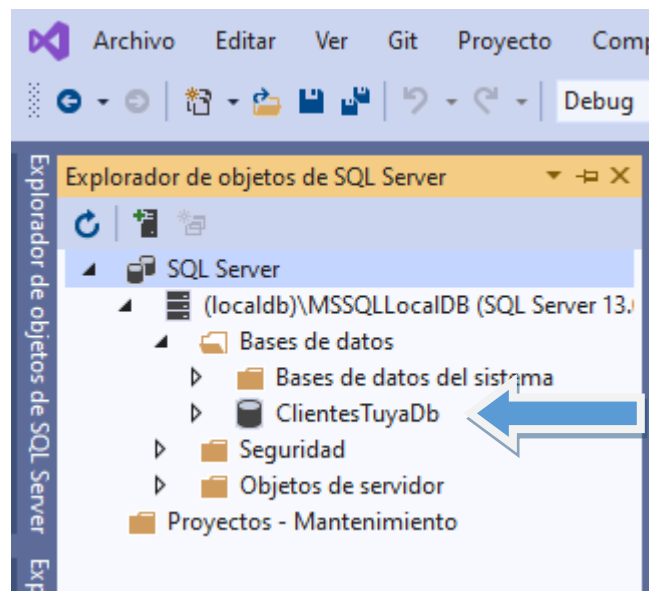
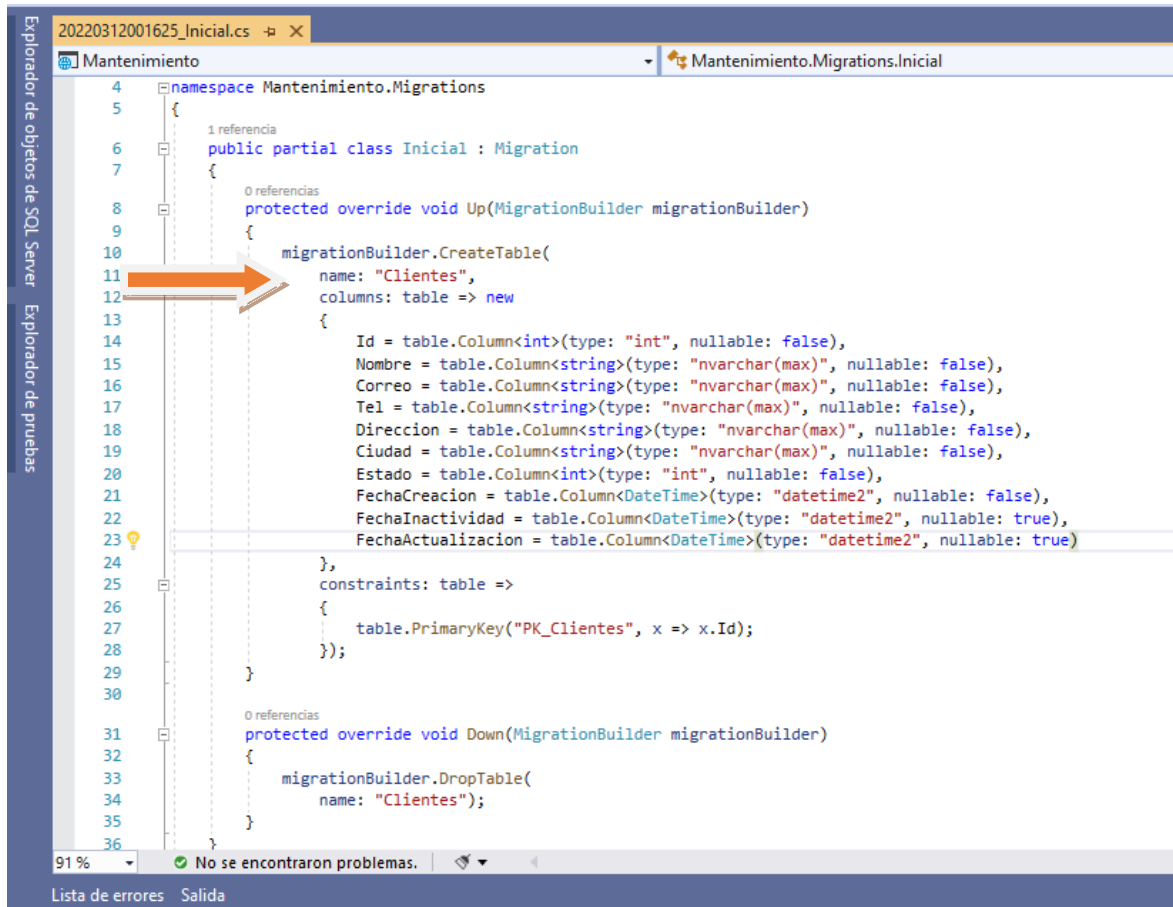


Gráfico 16. Creación de la base de datos ClientesTuyaDb en SQL Server

Tomada de: Hecho por el practicante, 2022

5.4. Creación de la tabla Clientes en la base de datos

En esta clase se llevó a cabo la creación de la única tabla que llevaría nuestra base de datos ClientesTuyaDb, le asignamos el nombre de Clientes a la tabla con sus respectivos atributos.



```

20220312001625_Inicial.cs
Maintenimiento
namespace Mantenimiento.Migrations
{
    1 referencia
    public partial class Inicial : Migration
    {
        0 referencias
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "Clientes",
                columns: table => new
                {
                    Id = table.Column<int>(type: "int", nullable: false),
                    Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Correo = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Tel = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Direccion = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Ciudad = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Estado = table.Column<int>(type: "int", nullable: false),
                    FechaCreacion = table.Column<DateTime>(type: "datetime2", nullable: false),
                    FechaInactividad = table.Column<DateTime>(type: "datetime2", nullable: true),
                    FechaActualizacion = table.Column<DateTime>(type: "datetime2", nullable: true)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Clientes", x => x.Id);
                });
        }
        0 referencias
        protected override void Down(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.DropTable(
                name: "Clientes");
        }
    }
}
91 % No se encontraron problemas.
Lista de errores Salida

```

Gráfico 17. Creación de la tabla Clientes en la base de datos ClientesTuyaDb

Tomada de: Hecho por el practicante, 2022

5.5. Creación de la Entidad principal de nuestra API

Esta actividad se llevó a cabo la creación de la clase principal de nuestro proyecto, dicha clase la llamamos Cliente, la cual contiene los atributos necesarios que debe tener un cliente a la hora de registrarse.


```

Cliente.cs
namespace Mantenimiento
{
    using System.Threading.Tasks;

    namespace Mantenimiento.Entities
    {
        1 referencia
        public class Cliente
        {
            12 [Key]
            13 [DatabaseGenerated(DatabaseGeneratedOption.None)]
            0 referencias
            public int Id { get; set; }

            16 [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
            0 referencias
            public string Nombre { get; set; }

            19 [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
            0 referencias
            public string Correo { get; set; }

            22 [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
            0 referencias
            public string Tel { get; set; }

            25 [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
            0 referencias
            public string Direccion { get; set; }

            28 [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
            0 referencias
            public string Ciudad { get; set; }

            31 [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
            0 referencias
            public int Estado { get; set; }

            33 public DateTime FechaCreacion { get; set; }
        }
    }
}

```

91 % No se encontraron problemas.

Gráfico 18. Creación de la entidad principal de nuestra API

Tomada de: Hecho por el practicante, 2022

```
33 public DateTime FechaCreacion { get; set; }  
34 0 referencias  
35 public DateTime? FechaInactividad { get; set; }  
36 0 referencias  
37 public DateTime? FechaActualizacion { get; set; }  
38 }
```

1 % No se encontraron problemas.

Gráfico 19. Creación de la entidad principal de nuestra API parte 2

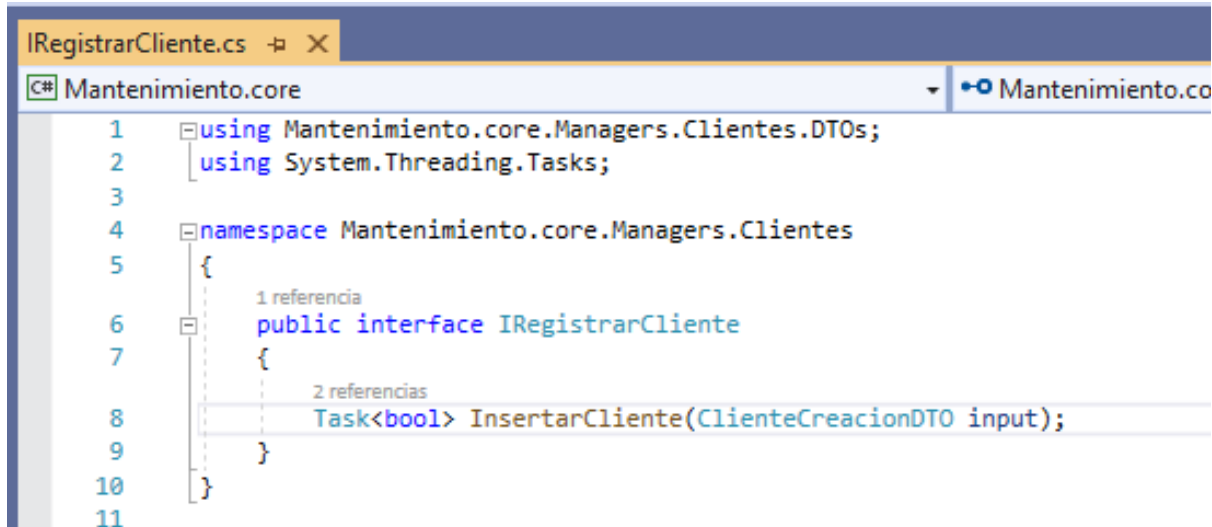
Tomada de: Hecho por el practicante, 2022

5.6. Creación de las funcionalidades requeridas en la API.

- Función insertar cliente con su respectiva interfaz:

➤ Interfaz Insertar Cliente:

Esta actividad consistió en la creación de la interface IRegistrarCliente, que usara nuestra clase ClienteCreacionDTO más adelante para registrar un nuevo cliente.



```
1 using Mantenimiento.core.Managers.Cientes.DTOs;
2 using System.Threading.Tasks;
3
4 namespace Mantenimiento.core.Managers.Cientes
5 {
6     1 referencia
7     public interface IRegistrarCliente
8     {
9         2 referencias
10         Task<bool> InsertarCliente(ClienteCreacionDTO input);
11     }
12 }
```

Gráfico 20. Interfaz IRegistrarCliente para insertar cliente nuevo

Tomada de: Hecho por el practicante, 2022

➤ **Clase donde se crea un cliente nuevo:**

En esta actividad, creamos la clase ClienteCreacionDTO y usamos la interfaz IRegistrarCliente, creada anteriormente para hacer el método con el cual crearemos el nuevo registro del cliente.

```
ClienteCreacionDTO.cs X IRegistrarCliente.cs
C# Mantenimiento.core Mantenimiento.cc

10
11 namespace Mantenimiento.core.Managers.Clientes.DTOs
12 {
13     6 referencias
14     public class ClienteCreacionDTO : IRegistrarCliente
15     {
16         private readonly MantenimientoContext context;
17         2 referencias
18         public int Id { get; set; }
19
20         [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
21         1 referencia
22         public string Nombre { get; set; }
23
24         [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
25         1 referencia
26         public string Correo { get; set; }
27
28         [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
29         1 referencia
30         public string Tel { get; set; }
31
32         [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
33         1 referencia
34         public string Direccion { get; set; }
35
36         [Required(ErrorMessage = "Campo {0} no puede ser vacio")]
37         1 referencia
38         public String Ciudad { get; set; }
39
40         1 referencia
41         public ClienteCreacionDTO(MantenimientoContext context)
42         {
43             this.context = context;
44         }
45     }
46 }
```

Gráfico 21. Clase ClienteCreacionDTO donde se crea un cliente nuevo

Tomada de: Hecho por el practicante, 2022

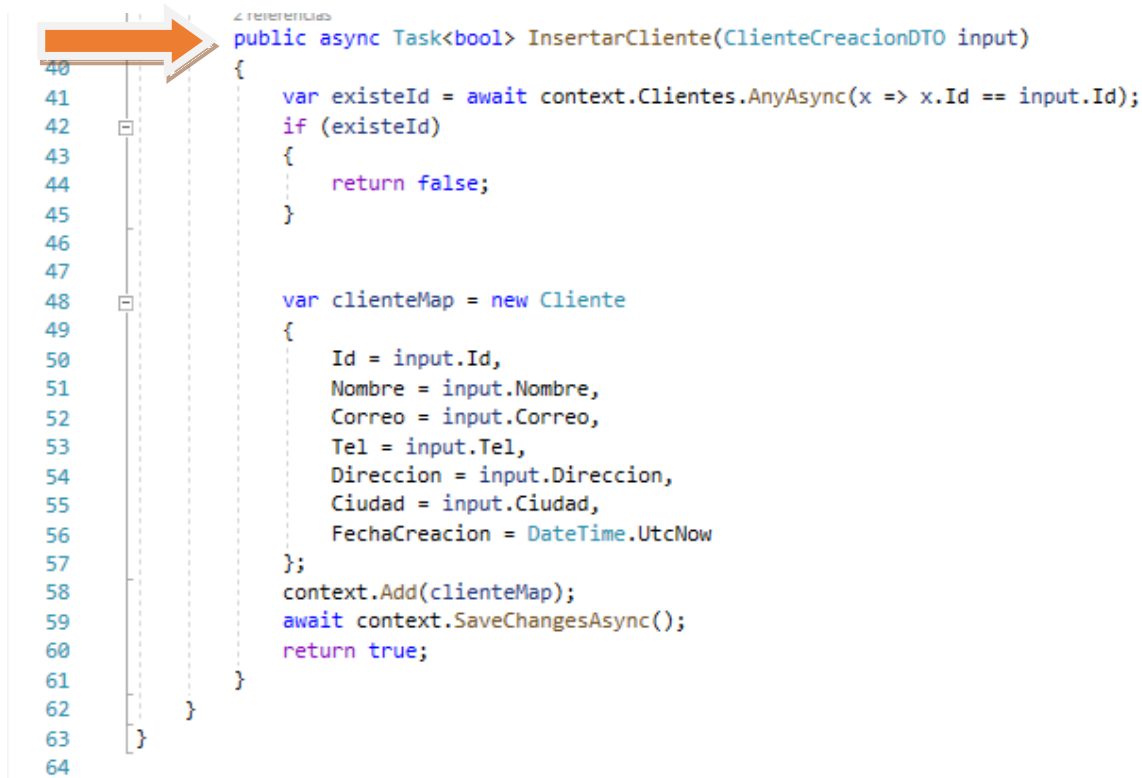


Gráfico 22. Clase ClienteCreacionDTO donde se crea un cliente nuevo parte 2

Tomada de: Hecho por el practicante, 2022

- **Función listar clientes por su número de documento con su respectiva interfaz:**

➤ **Interfaz para listar un cliente por su número de documento:**

En esta actividad, se creó la interfaz `IListarDoc`, que usara nuestra clase `ClienteListarDocDTO` más adelante, para buscar un cliente por medio de su número de documento ingresado previamente

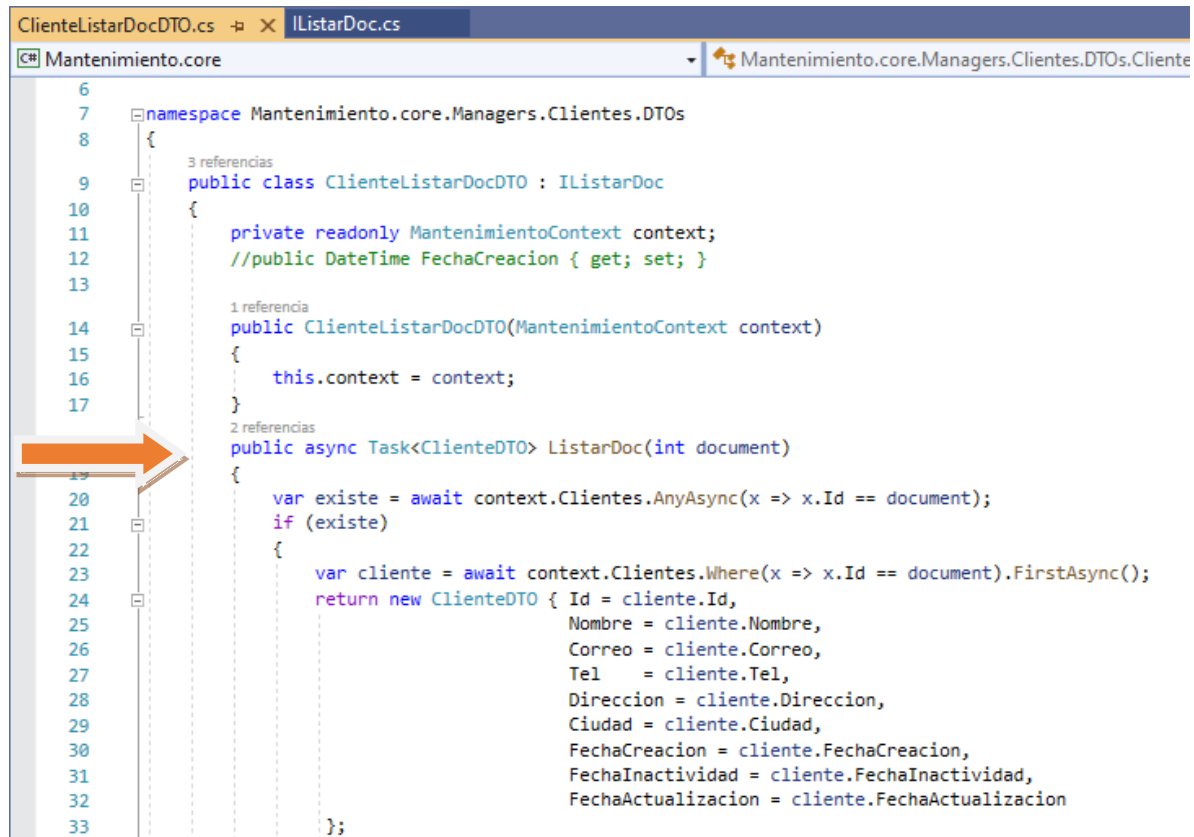
```
ClienteListarDocDTO.cs  IListarDoc.cs  X
C# Mantenimiento.core
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Mantenimiento.core.Managers.Cientes.DTOs
8  {
9      1 referencia
10     interface IListarDoc
11     {
12         2 referencias
13         Task<ClienteDTO> ListarDoc(int document);
14     }
15 }
```

Gráfico 23. Interfaz IListarDoc para buscar un cliente por su número de documento

Tomada de: Hecho por el practicante, 2022

➤ **Clase donde se lista un cliente por su número de documento:**

En esta actividad, creamos la clase ClienteListarDocDTO y usamos la interfaz IListarDoc, creada anteriormente para hacer el método con el cual buscaremos con el número de documento el registro del cliente.



```

1  ClienteListarDocDTO.cs  x  IListarDoc.cs
2  Mantenimiento.core
3  namespace Mantenimiento.core.Managers.Cientes.DTOs
4  {
5      3 referencias
6      public class ClientelListarDocDTO : IListarDoc
7      {
8          private readonly MantenimientoContext context;
9          //public DateTime FechaCreacion { get; set; }
10
11          1 referencia
12          public ClientelListarDocDTO(MantenimientoContext context)
13          {
14              this.context = context;
15          }
16
17          2 referencias
18          public async Task<ClienteDTO> ListarDoc(int document)
19          {
20              var existe = await context.Cientes.AnyAsync(x => x.Id == document);
21              if (existe)
22              {
23                  var cliente = await context.Cientes.Where(x => x.Id == document).FirstAsync();
24                  return new ClienteDTO { Id = cliente.Id,
25                                          Nombre = cliente.Nombre,
26                                          Correo = cliente.Correo,
27                                          Tel = cliente.Tel,
28                                          Direccion = cliente.Direccion,
29                                          Ciudad = cliente.Ciudad,
30                                          FechaCreacion = cliente.FechaCreacion,
31                                          FechaInactividad = cliente.FechaInactividad,
32                                          FechaActualizacion = cliente.FechaActualizacion
33              };
34          }
35      }
36  }
    
```

Gráfico 24. Clase ClienteListarDocDTO donde se lista un cliente por su número de documento

Tomada de: Hecho por el practicante, 2022

```
35  
36  
37  
38  
39  
40  
41  
42  
  
else  
{  
    throw new ArgumentException("El documento no existe");  
}
```

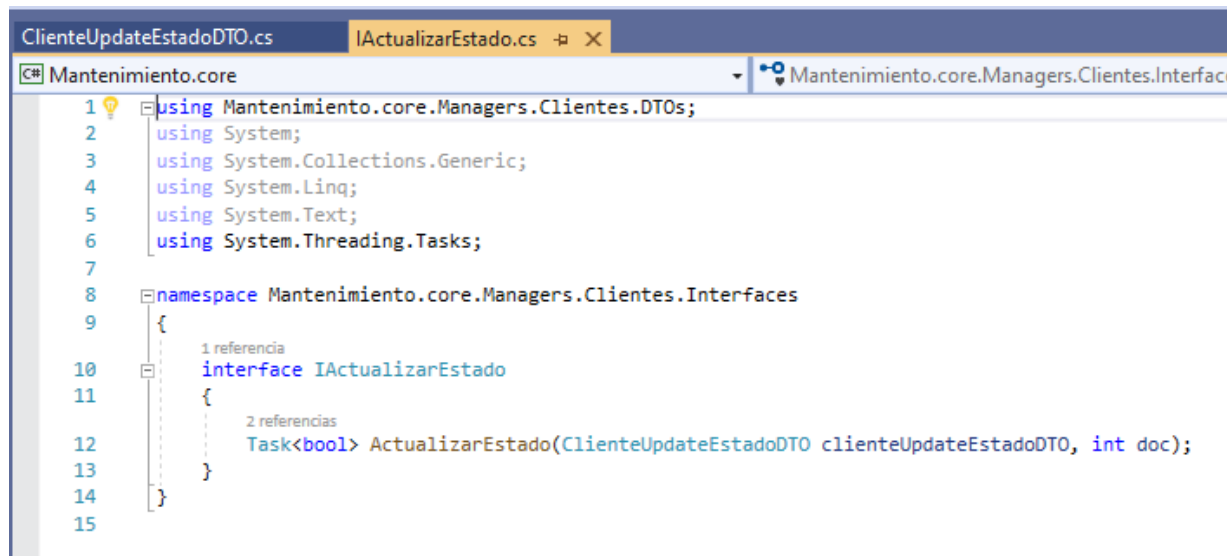
Gráfico 25. Clase ClienteListarDocDTO donde se lista un cliente por su número de documento parte 2

Tomada de: Hecho por el practicante, 2022

- **Función inactivar cliente por su número de documento con su respectiva interfaz:**

➤ **Interfaz para inactivar cliente por su número de documento:**

En esta actividad, se creó la interfaz IActualizarEstado, que usara nuestra clase ClienteUpdateEstadoDTO más adelante, para cambiar el estado ya sea activo o inactivo de un cliente por medio de su número de documento ingresado previamente.



```

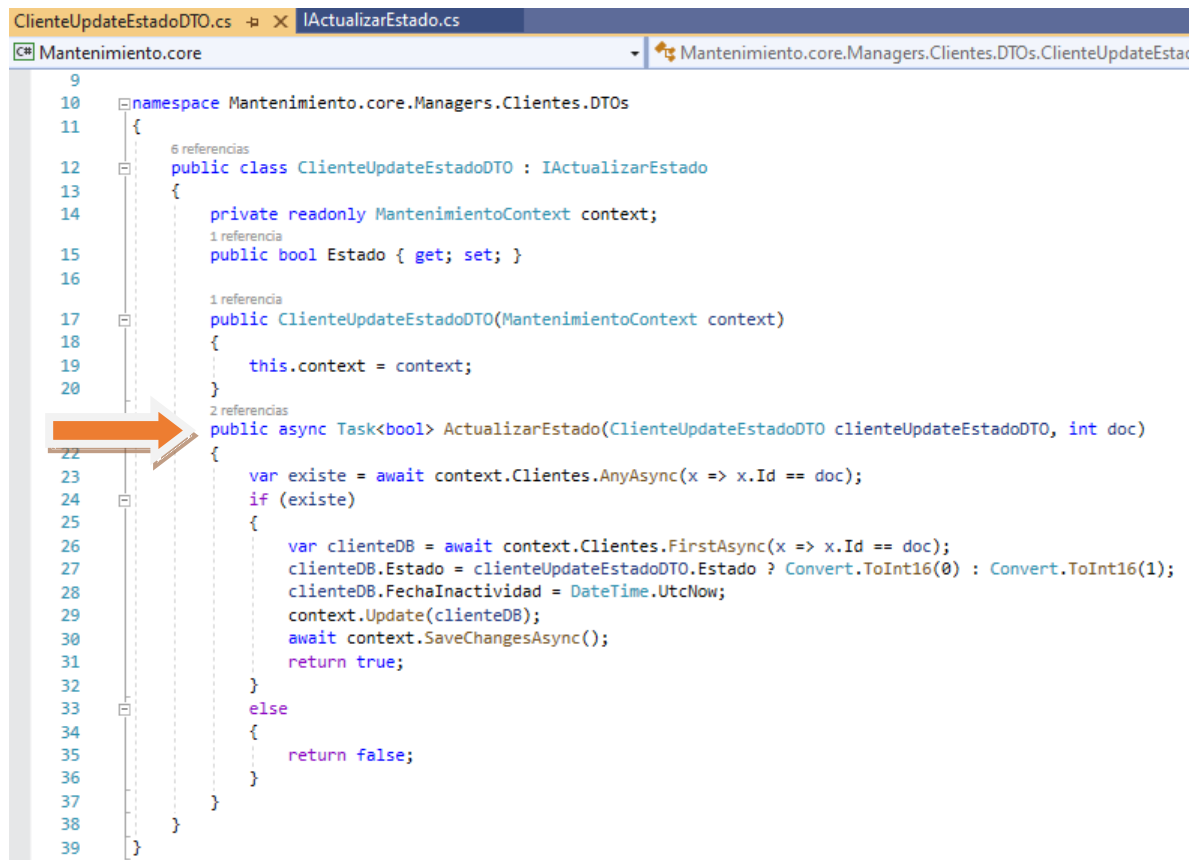
1  using Mantenimiento.core.Managers.Clientes.DTOs;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace Mantenimiento.core.Managers.Clientes.Interfaces
9  {
10     1 referencia
11     interface IActualizarEstado
12     {
13         2 referencias
14         Task<bool> ActualizarEstado(ClienteUpdateEstadoDTO clienteUpdateEstadoDTO, int doc);
15     }
16 }
    
```

Gráfico 26. Interfaz IActualizarEstado para inactivar cliente por su número de documento

Tomada de: Hecho por el practicante, 2022

➤ Clase para inactivar un cliente por su número de documento:

En esta actividad, creamos la clase ClienteUpdateEstadoDTO y usamos la interfaz IActualizarEstado, creada anteriormente para hacer el método con el cual activaremos o inactivaremos a un cliente por medio de su número de documento.



```

9
10 namespace Mantenimiento.core.Managers.Cientes.DTOs
11 {
12     6 referencias
13     public class ClienteUpdateEstadoDTO : IActualizarEstado
14     {
15         private readonly MantenimientoContext context;
16         1 referencia
17         public bool Estado { get; set; }
18
19         1 referencia
20         public ClienteUpdateEstadoDTO(MantenimientoContext context)
21         {
22             this.context = context;
23         }
24         2 referencias
25         public async Task<bool> ActualizarEstado(ClienteUpdateEstadoDTO clienteUpdateEstadoDTO, int doc)
26         {
27             var existe = await context.Cientes.AnyAsync(x => x.Id == doc);
28             if (existe)
29             {
30                 var clienteDB = await context.Cientes.FirstAsync(x => x.Id == doc);
31                 clienteDB.Estado = clienteUpdateEstadoDTO.Estado ? Convert.ToInt16(0) : Convert.ToInt16(1);
32                 clienteDB.FechaInactividad = DateTime.UtcNow;
33                 context.Update(clienteDB);
34                 await context.SaveChangesAsync();
35                 return true;
36             }
37             else
38             {
39                 return false;
40             }
41         }
42     }
43 }
    
```

Gráfico 27. Clase ClienteUpdateEstadoDTO para inactivar un cliente por su número de documento

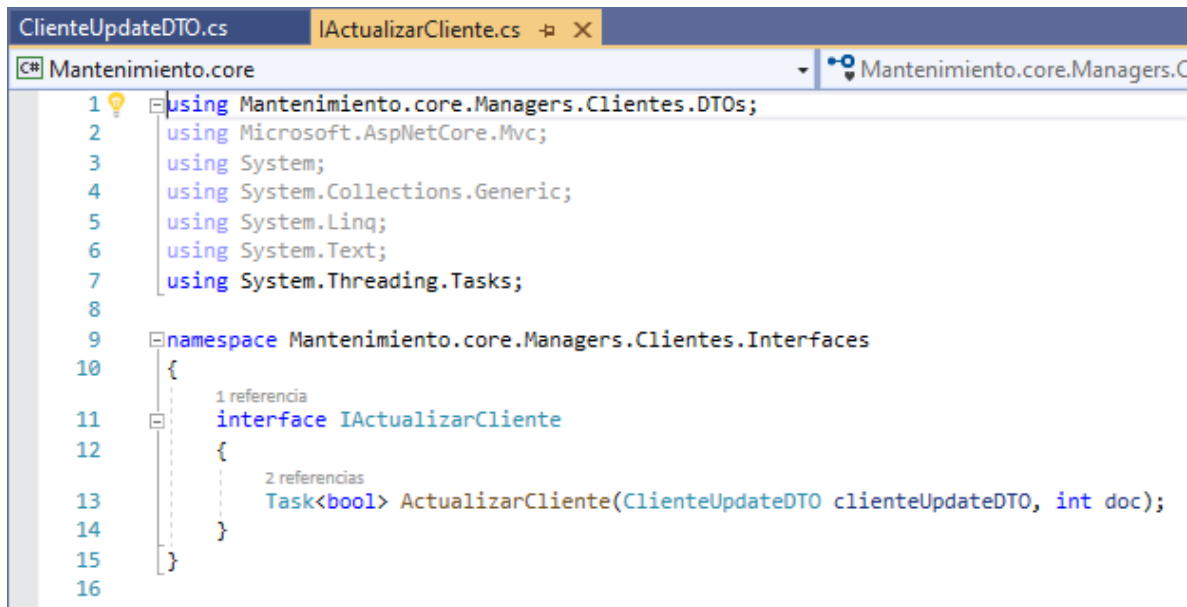
Tomada de: Hecho por el practicante, 2022

- **Función actualizar datos personales del cliente por su número de documento con su respectiva interfaz:**

- **Interfaz para actualizar los datos de un cliente por su número de documento:**

En esta actividad, se creó la interfaz IActualizarCliente, que usara nuestra clase ClienteUpdateDTO más adelante, para actualizar todos los datos.

excepto el número de documento de un cliente previamente registrado en la base de datos.



```

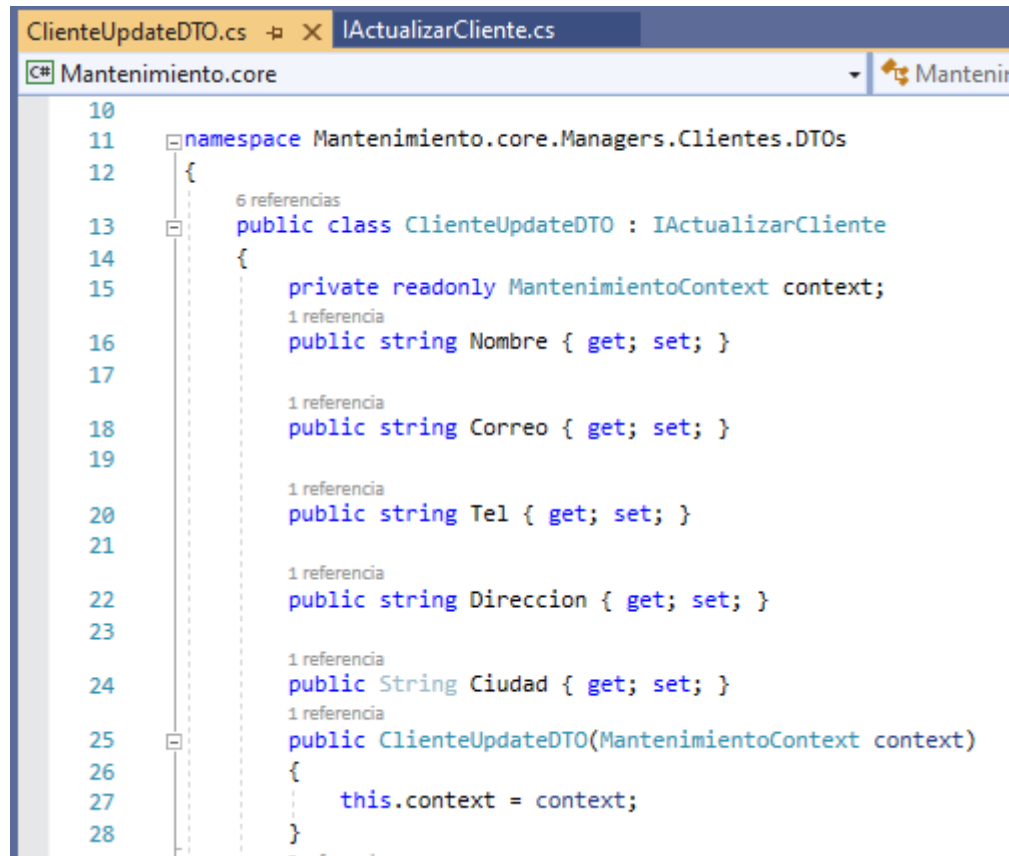
1  using Mantenimiento.core.Managers.Cientes.DTOs;
2  using Microsoft.AspNetCore.Mvc;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace Mantenimiento.core.Managers.Cientes.Interfaces
10 {
11     1 referencia
12     interface IActualizarCliente
13     {
14         2 referencias
15         Task<bool> ActualizarCliente(ClienteUpdateDTO clienteUpdateDTO, int doc);
16     }
17 }
    
```

Gráfico 28. Actualización de los datos del cliente por su número de documento por medio la interfaz IActualizarCliente.cs

Tomada de: Hecho por el practicante, 2022

➤ Clase para actualizar los datos de un cliente por su número de documento:

En esta actividad, creamos la clase ClienteUpdateDTO y usamos la interfaz IActualizarCliente, creada anteriormente para hacer el método con el cual actualizaremos los datos excepto el número de documento a un cliente por registrado previamente en la base de datos.



```

10
11 namespace Mantenimiento.core.Managers.Cientes.DTOs
12 {
13     6 referencias
14     public class ClienteUpdateDTO : IActualizarCliente
15     {
16         private readonly MantenimientoContext context;
17         1 referencia
18         public string Nombre { get; set; }
19
20         1 referencia
21         public string Correo { get; set; }
22
23         1 referencia
24         public string Tel { get; set; }
25
26         1 referencia
27         public string Direccion { get; set; }
28
29         1 referencia
30         public String Ciudad { get; set; }
31
32         1 referencia
33         public ClienteUpdateDTO(MantenimientoContext context)
34         {
35             this.context = context;
36         }
37     }
38 }
    
```

Gráfico 29. Actualización de los datos de un cliente por su número de documento con la clase ClienteUpdateDTO.cs

Tomada de: Hecho por el practicante, 2022

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías

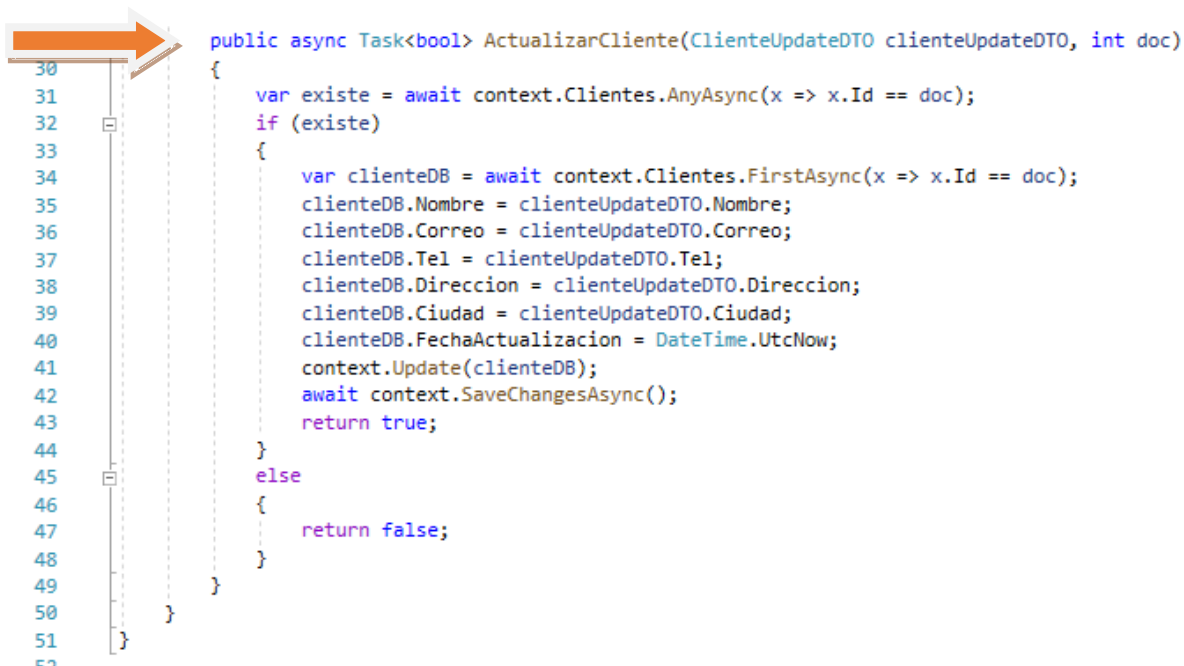
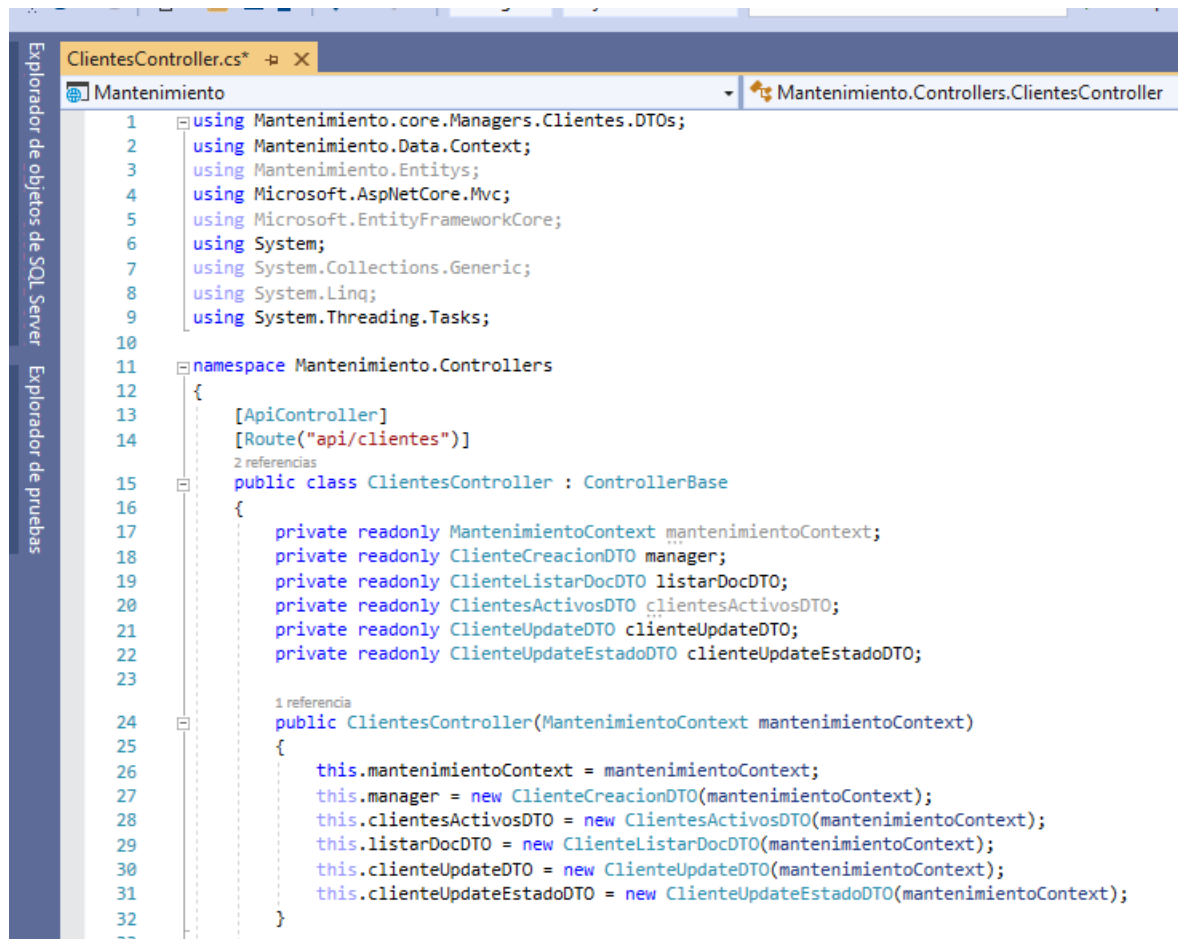


Gráfico 30. Actualización de los datos de un cliente por su número de documento con la clase ClienteUpdateDTO.cs parte 2

Tomada de: Hecho por el practicante, 2022

5.7. Clase para controlar todas las funciones de la API

En esta clase vamos a llamar todas las funciones creadas anteriormente mediante y usaremos los protocolos HTML para la ejecución de la API.



```

1  using Mantenimiento.core.Managers.Cientes.DTOS;
2  using Mantenimiento.Data.Context;
3  using Mantenimiento.Entities;
4  using Microsoft.AspNetCore.Mvc;
5  using Microsoft.EntityFrameworkCore;
6  using System;
7  using System.Collections.Generic;
8  using System.Linq;
9  using System.Threading.Tasks;
10
11 namespace Mantenimiento.Controllers
12 {
13     [ApiController]
14     [Route("api/clientes")]
15     2 referencias
16     public class CientesController : ControllerBase
17     {
18         private readonly MantenimientoContext mantenimientoContext;
19         private readonly ClienteCreacionDTO manager;
20         private readonly ClienteListarDocDTO listarDocDTO;
21         private readonly ClientesActivosDTO clientesActivosDTO;
22         private readonly ClienteUpdateDTO clienteUpdateDTO;
23         private readonly ClienteUpdateEstadoDTO clienteUpdateEstadoDTO;
24
25         1 referencia
26         public CientesController(MantenimientoContext mantenimientoContext)
27         {
28             this.mantenimientoContext = mantenimientoContext;
29             this.manager = new ClienteCreacionDTO(mantenimientoContext);
30             this.clientesActivosDTO = new ClientesActivosDTO(mantenimientoContext);
31             this.listarDocDTO = new ClienteListarDocDTO(mantenimientoContext);
32             this.clienteUpdateDTO = new ClienteUpdateDTO(mantenimientoContext);
33             this.clienteUpdateEstadoDTO = new ClienteUpdateEstadoDTO(mantenimientoContext);
34         }
35     }
36 }
    
```

Gráfico 31. Controlador de las funciones de la API

Tomada de: Hecho por el practicante, 2022

➤ Método HttpPost para registrar cliente

En esta actividad creamos el método HttpPost en nuestra clase CientesController, este método se utilizará para registrar clientes en la base de datos.

```
//Insertar Cliente
[HttpPost]
0 referencias
public async Task<ActionResult> Post(ClienteCreacionDTO clienteCreacionDTO)
{
    var result = await manager.InsertarCliente(clienteCreacionDTO);

    if (result)
    {
        return Ok("Cliente creado");
    }
    else
    {
        return BadRequest("Ya existe un cliente registrado con este numero de cedula");
    }
}
```

Gráfico 32. Método Post para registrar clientes a la base de datos

Tomada de: Hecho por el practicante, 2022

➤ Método HttpGet para buscar un cliente

En esta actividad creamos el método HttpGet en nuestra clase ClientesController, este método se utilizará para Buscar un cliente en la base de datos pasando como parámetro el número de documento registrado anteriormente.



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



```
//Listar por documento
[HttpGet("{doc:int}")]
1 referencia
public async Task<ActionResult<ClienteDTO>> GetDocument(int doc)
{
    try
    {
        var result = await listarDocDTO.ListarDoc(doc);
        return Ok(result);
    }
    catch (ArgumentException e)
    {
        return BadRequest(e.Message);
    }
}
```

Gráfico 33. Método Get para buscar un cliente en la base de datos

Tomada de: Hecho por el practicante, 2022

➤ Método HttpPut para inactivar un cliente

En esta actividad creamos el método HttpPut en nuestra clase ClientesController, este método se utilizará para inactivar o activar un cliente en la base de datos pasando como parámetro que estado desea colocarle y el número de documento registrado anteriormente.



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co


```
//Inactivar cliente  
[HttpPut("estado/{doc:int}")]  
0 referencias  
public async Task<ActionResult> Put(ClienteUpdateEstadoDTO clienteUpdateEstadoDT01, int doc)  
{  
    var result = await clienteUpdateEstadoDTO.ActualizarEstado(clienteUpdateEstadoDT01, doc);  
    if (result)  
    {  
        return Ok("Estado del Cliente Actualizado");  
    }  
    else  
    {  
        return BadRequest("Problemas al actualizar estado del cliente");  
    }  
}
```

Gráfico 34. Método Put para cambiar el estado de un cliente en la base de datos

Tomada de: Hecho por el practicante, 2022

➤ Método HttpPut para actualizar un cliente

En esta actividad creamos el método HttpPut en nuestra clase ClientesController, este método se utilizará para actualizar los datos de un cliente en la base de datos excepto el número de documento, esta acción se realiza pasando como parámetro el número de documento del cliente a actualizar registrado anteriormente.

```
//Actualizar cliente
[HttpPut("{doc:int}")]
0 referencias
public async Task<ActionResult> Put(ClienteUpdateDTO clienteUpdateDT01, int doc)
{
    var result = await clienteUpdateDT01.ActualizarCliente(clienteUpdateDT01, doc);
    if (result)
    {
        return Ok("Cliente Actualizado");
    }
    else
    {
        return BadRequest("Problemas al actualizar cliente");
    }
}
```

Gráfico 35. Método Put para actualizar los datos de un cliente en la base de datos

Tomada de: Hecho por el practicante, 2022

5.8. Reporte final sobre el estado de la API.

Por medio de una reunión con la líder de equipo Marcela Restrepo López en la plataforma Teams, entregue la creación de la API para su revisión, para así proceder con la ejecución de la misma.

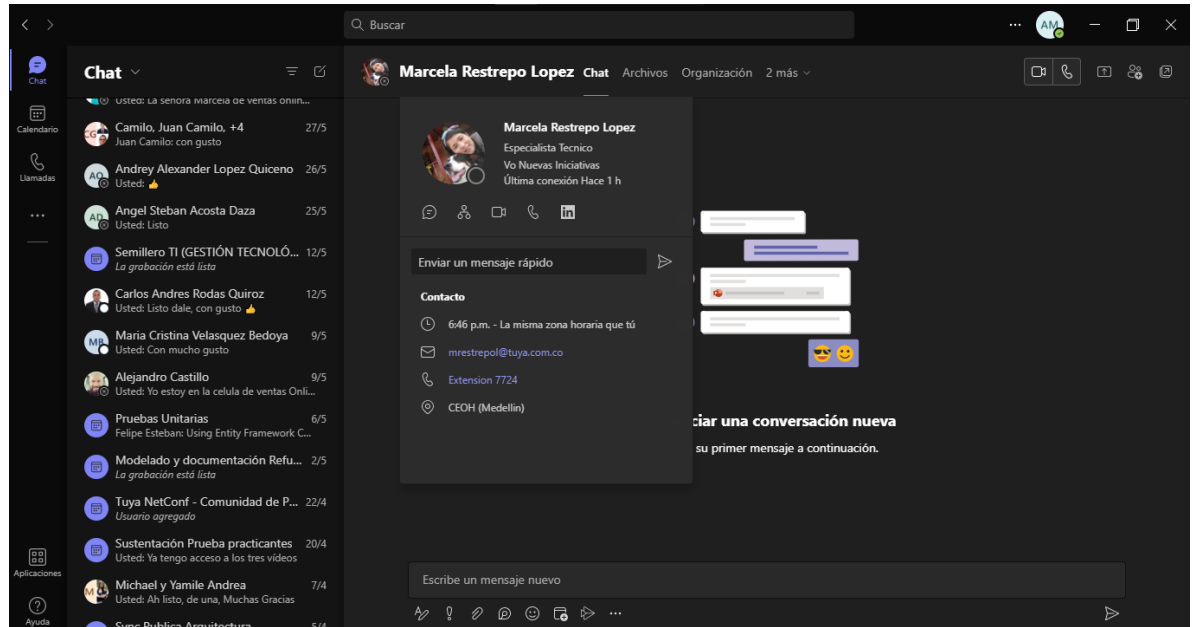


Gráfico 36. Reunión de reporte final con la líder Marcela Restrepo López

Tomada de: Hecho por el practicante, 2022

5.9. Ejecución de la API.

En esta etapa, ya nuestra API esta lista para ser ejecutada y así verificar su funcionamiento.

En la siguiente imagen podrán observar cada uno de los métodos HTTP creados anteriormente en nuestra clase ClientesController.

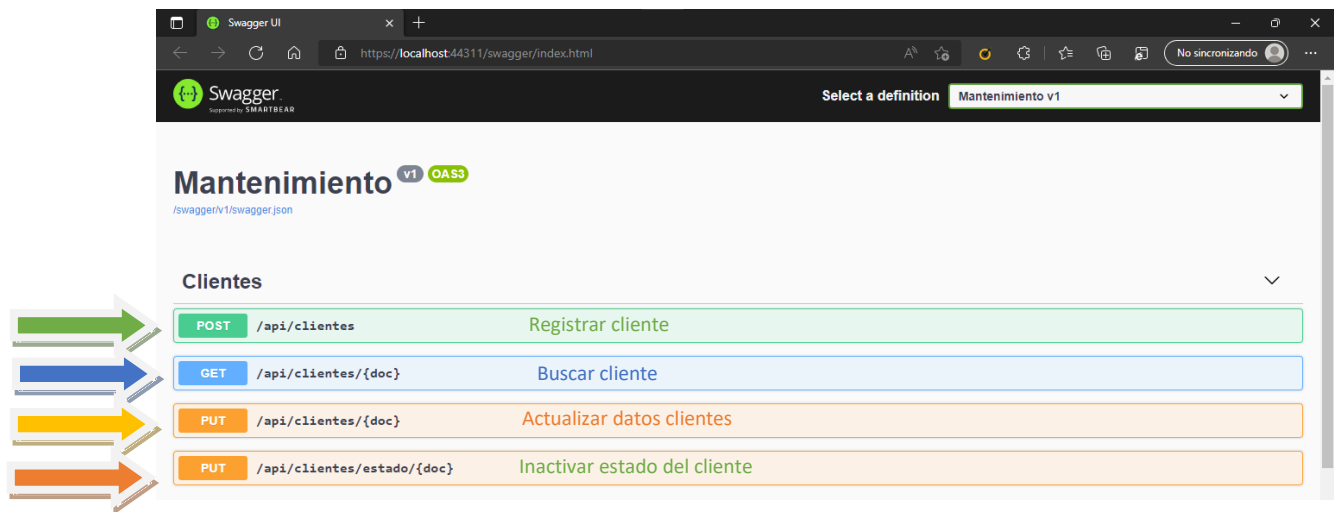


Gráfico 37. Ejecución de la API

Tomada de: Hecho por el practicante, 2022



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



6. PROPUESTA DE MEJORAMIENTO

Se trabajo en el departamento de ventas Online en busca de brindar alternativas en el manejo de los datos de los clientes, a través de la creación de una API de prueba para que a futuro la empresa pueda tener mejores opciones en el manejo de datos de los clientes.

7. APORTES DEL ESTUDIANTE

Con el desarrollo de la API se lograron alcanzar los objetivos y expectativas, cumpliendo con todos los requisitos establecidos por la compañía. Destacando los siguientes:

- El agilizar los procesos de registro y manejo de clientes en la célula de Ventas Online.
- El manejo de la metodología Scrum como línea de trabajo.
- La creación de una nueva base de datos para la compañía.
- El mejoramiento de la protección a la información de los nuevos clientes de la compañía.

Además de mejorar una parte de la célula de ventas online de la compañía, también destaco el compromiso de la Compañía TUYA S.A hacia sus clientes, puesto que la creación de la API mejora las condiciones para adquirir cualquier producto de la compañía.



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11. INFORME FINAL PRÁCTICA EMPRESARIAL Comité de Acreditación y Currículo Facultad de Ingenierías



8. CONCLUSIONES

TUYA S.A es una compañía abierta a brindar la experiencia laboral innovadora para muchos jóvenes. Desde las labores asignadas y ejecutadas se pusieron a prueba muchos conocimientos, pero a su vez se adquirieron nuevas destrezas.

Cuando se trabajan en procesos de creación y desarrollo es necesario conocer cuales son los objetivos a alcanzar, porque de esta manera se organiza mejor las formas y los esquemas de trabajo.

Todo el transcurso de las practicas fue muy beneficioso aun cuando estas se ejecutaron desde casa, esto a causa de la prevención al problema de salud publica que atraviesa el mundo. Agradezco a la Compañía Financiera TUYA S.A por la oportunidad que me brindaron al trabajar junto a ellos.



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11. INFORME FINAL PRÁCTICA EMPRESARIAL Comité de Acreditación y Currículo Facultad de Ingenierías



9. RECOMENDACIONES

Con base a las actividades realizadas en la Compañía TUYA S.A, en la célula de ventas online se hace necesario dejar algunas recomendaciones para seguir en busca de mejorar.

- Asignar a un practicante o empleado con conocimientos del lenguaje de programación React.js, con la intención de complementar la API con la creación de un frontend, esto con la intención que el manejo de la misma sea mas amigable e intuitiva.
- Encargar el mantenimiento de la API a un empleado o futuro practicante, para garantizar el buen funcionamiento de la misma todo el tiempo posible.



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

PLANTILLA 11.

INFORME FINAL PRÁCTICA EMPRESARIAL

Comité de Acreditación y Currículo Facultad de Ingenierías



10. BIBLIOGRAFÍA

- Tuya. (s. f.). Compañía Tuya. Recuperado 3 de febrero de 2022, de <https://www.tuya.com.co/nuestra-compania>
- Cardozo, L. (2021, 6 diciembre). *API: ¿Qué es API, ejemplos y su importancia!* Zenvia. <https://www.zenvia.com/es/blog/que-es-api/>
- SAS. (S.f). Caso de éxito TUYA. SAS. Recuperado junio 26, 2022, de <https://www.sas.com/content/dam/SAS/documents/partner-collateral/es/tuya-customer-story.pdf>
- Schwaber, K. and Jeff Sutherland, J., 2020. *La Guía Scrum*. Scrumguides.org. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- Red Hat. (2017). *¿Qué es una API?* Red Hat. Recuperado June 26, 2022, de <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
- Ferrara, P. (2014, 30 junio). *Controlador Web API de ASP.NET MVC*. Solvetic. <https://www.solvetic.com/tutoriales/article/859-controlador-web-api-de-aspnet-mvc/>
- *Métodos de petición HTTP - HTTP / MDN*. (2021, 22 junio). Developer Mozilla. <https://developer.mozilla.org/es/docs/Web/HTTP/Methods>
- Cruz-Chávez, M. A. (2011). Bases de Datos, Conceptos y sus Características. Centro de Investigaciones en Ingeniería y Ciencias Aplicadas, Universidad Autónoma del Estado de Morelos. <http://www.gridmorelos.uaem.mx/~mcruz/cursos/miic/bd1.pdf>
- Stefaniak, P. (2019, 10 septiembre). *¿Qué es Backend y Frontend?* Descubre Comunicación. <https://descubrecomunicacion.com/que-es-backend-y-frontend/>



Por una universidad con calidad, moderna e incluyente

Carrera 6ª. No. 76-103 Montería NIT. 891080031-3 - Teléfono: 7860300 - 7860920 www.unicordoba.edu.co